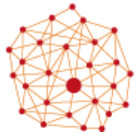


LocalSolver

Solving routing and scheduling problems using LocalSolver

Set-based modeling in LocalSolver 6.5

www.localsolver.com

Frédéric GARDI
Co-Founder & Managing Partner
fgardi@localsolver.com

Who we are

Bouygues, one of the French largest corporation, €33 bn in revenues

http://www.bouygues.com

Innovation24

Operations Research subsidiary of Bouygues
20 years of practice and research

http://www.innovation24.fr

LocalSolver

Mathematical optimization solver developed by Innovation 24

http://www.localsolver.com

Swiss Army Knife for math optimization

All-terrain

Discrete & Numerical

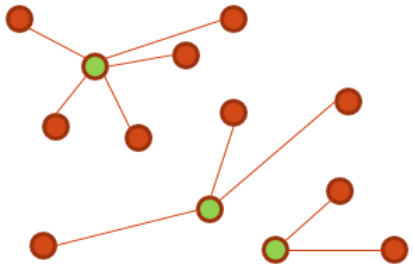
Fast & Scalable

But… ≠ MIP ≠CP

Clients

- Construction
- Medias & Advertising
- Telco & Retail
- Large Industry
- Energy
- Banking & Finance
- Transportation
- Logistics
- Food & Agribusiness
- Aerospace & Defense
- IT Services

Facility location



Select a subset P among N points minimizing the sum of distances from each point in N to the nearest point in P

```
function model() {
  x[1..N] <- bool() ; // decisions: point i belongs to P if x[i] = 1
  constraint sum[i in 1..N]( x[i] ) == P ; // constraint: P points selected among N
  minDist[i in 1..N] <- min[j in 1..N]( x[j] ? Dist[i][j] : InfiniteDist ) ; // expressions: distance to the nearest point in P
  minimize sum[i in 1..N]( minDist[i] ) ; // objective: to minimize the sum of distances
}
```

Nothing else to write: "model & run" approach

- Straightforward, natural mathematical model
- Direct resolution: no tuning

Mathematical operators

Decisional	Arithmetical			Logical	Relational	Set
bool	sum	sub	prod	not	eq	count
float	min	max	abs	and	neq	indexof
int	div	mod	sqrt	or	geq	partition
list	log	exp	pow	xor	leq	disjoint
	cos	sin	tan	iif	gt	
	floor	ceil	round	array+at	lt	
	dist	scalar		piecewise		

+ operator call : to call an external native function
which can be used to implement your own (black-box) operator

Motivations

Modeling approaches for
the Traveling Salesman Problem

MIP modeling

Classical formulation [Dantzig, Fulkerson & Johnson, 1954]

- Variable for each edge → linear number of variables
- Exponential number of constraints → iterative subtour elimination scheme

Flow-based formulation [Gavish & Graves, 1978]

- Quadratic number of variables
- Quadratic number of constraints

In Orman & Williams (2006): "A survey of different integer programming formulations of the TSP"

Natural modeling

TSP as a permutation

The Traveling Salesman Problem (TSP)

TSP: Given a list of cities and their pairwise distances, find a shortest possible tour that visits each city exactly once.

Objective: find a permutation $a_1,\ldots,a_n$ of the cities that minimizes

$$d(a_1, a_2) + d(a_2, a_3) + \ldots + d(a_{n-1}, a_n) + d(a_n, a_1)$$

where $d(i, j)$ is the distance between cities i and j

An optimal TSP tour through Germany's 15 largest cities

In Rosen (2012): "Discrete mathematics and its applications"

Set-based modeling

Innovative modeling concepts
for routing & scheduling problems

List variables

Structured decisional operator list(n)

- Order a subset of values in domain {0, …, n-1}
- Each value is unique in the list

Classical operators to interact with "list"

- count(u): number of values selected in the list
- at(u,i) or u[i]: value at index i in the list
- indexOf(u,v): index of value v in the list
- disjoint(u1, u2, …, uk): true if u1, u2, …, uk are pairwise disjoint
- partition(u1, u2, …, uk): true if u1, u2, …, uk induce a partition of {0, …, n-1}

Traveling salesman

```
function model() {
  x <- list(N) ; // order n cities {0, ..., n-1} to visit
  constraint count(x) == N; // exactly n cities to visit
  minimize sum[i in 1..N-1]( Dist[ x[i-1] ][ x[i] ] )
              + Dist[ x[N-1] ][ x[0] ] ; // minimize sum of traveled distances
}
```

TSP: Given a list of cities and their pairwise distances, find a shortest possible tour that visits each city exactly once.

Objective: find a permutation $a_1,\ldots,a_n$ of the cities that minimizes

$$d(a_1, a_2) + d(a_2, a_3) + \ldots + d(a_{n-1}, a_n) + d(a_n, a_1)$$

where $d(i, j)$ is the distance between cities i and j

An optimal TSP tour through Germany's 15 largest cities

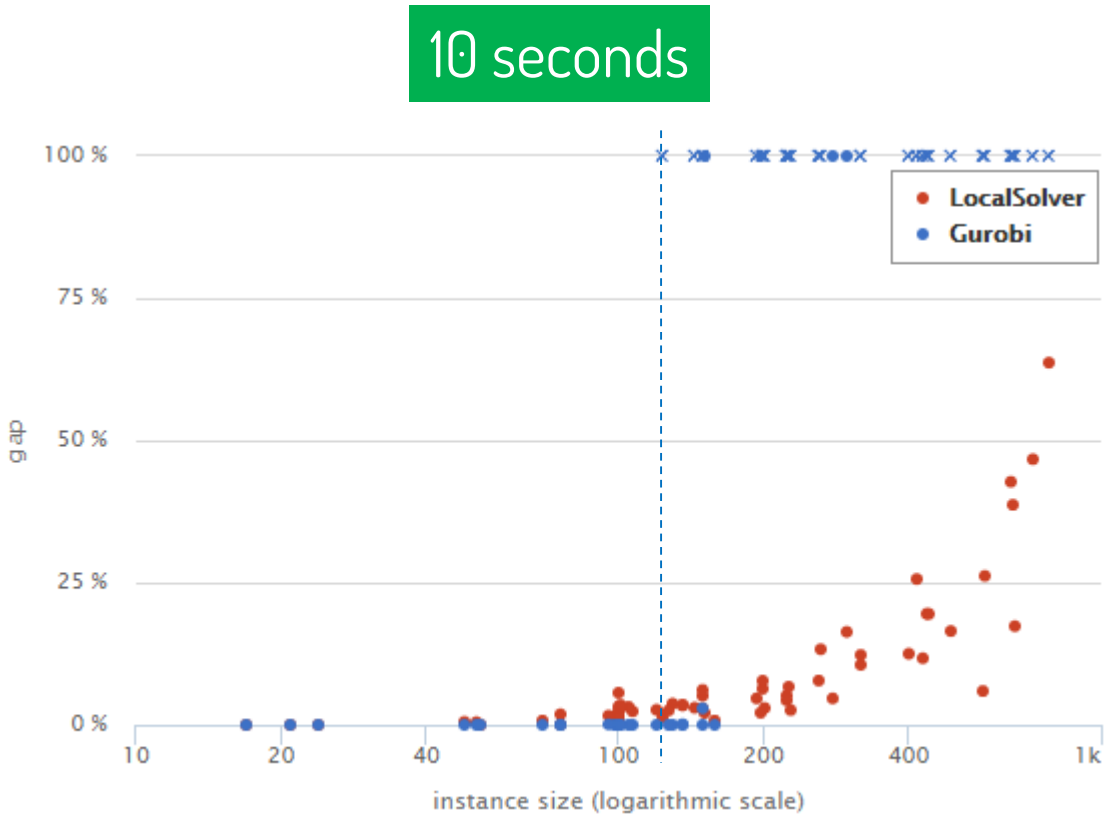
Comparison with MIP

TSP Lib instances

- Symmetric
- Size: 21 to 800 cities

x ⇔ no solution found

10 seconds



Gurobi 6.0

LocalSolver 6.0

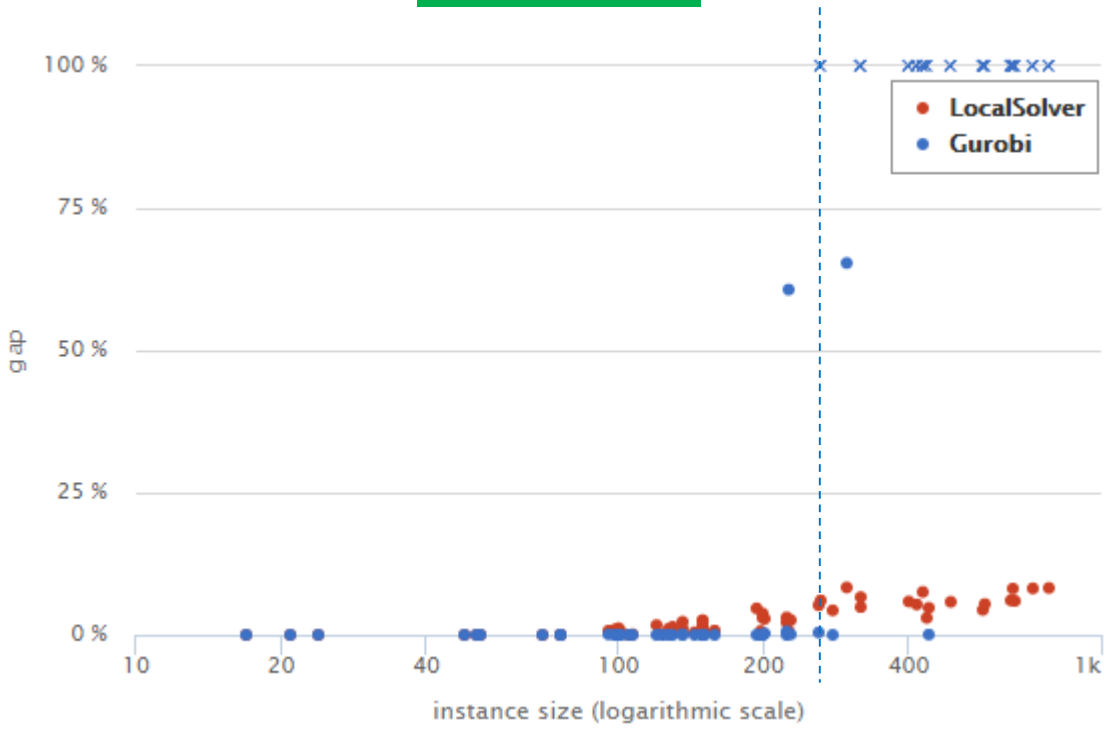
Comparison with MIP

TSP Lib instances

- Symmetric
- Size: 21 to 800 cities

x ⇔ no solution found

3 minutes



Gurobi 6.0

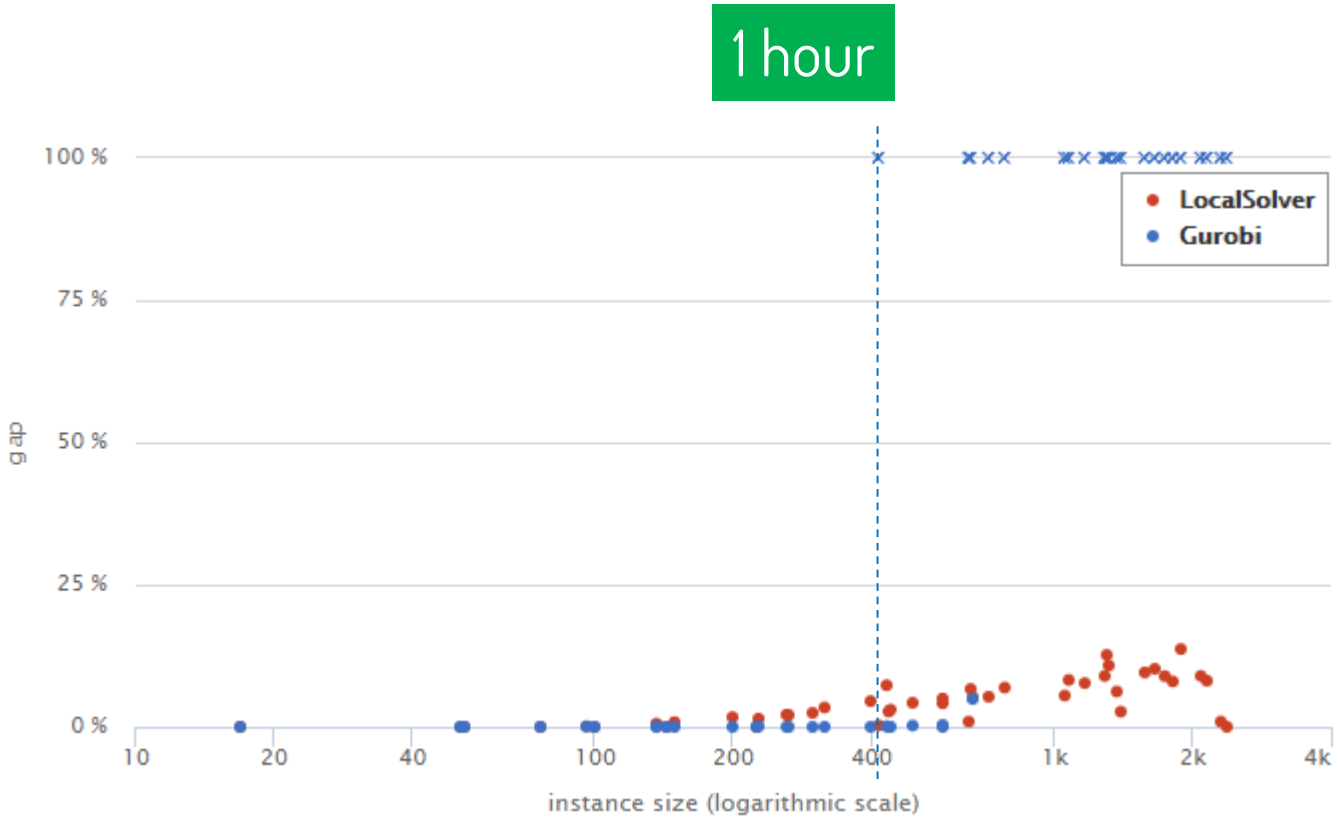
LocalSolver 6.0

Comparison with MIP

TSP Lib instances

- Symmetric
- Size: 21 to 800 cities (and more)

x ⇔ no solution found



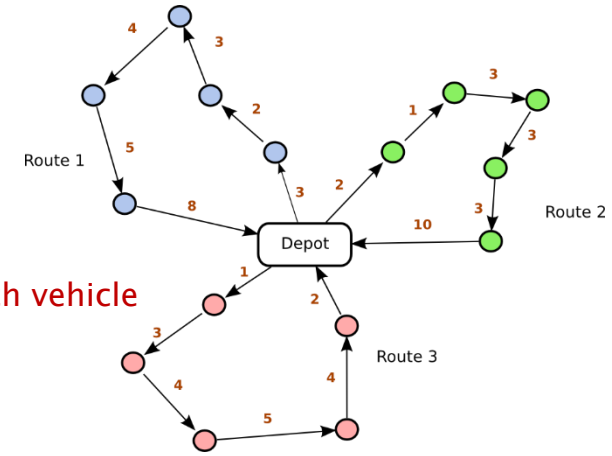
Gurobi 6.0

LocalSolver 6.0

Vehicle routing

Find the shortest set of routes for a fleet of K vehicles in order to deliver to a given set of N customers

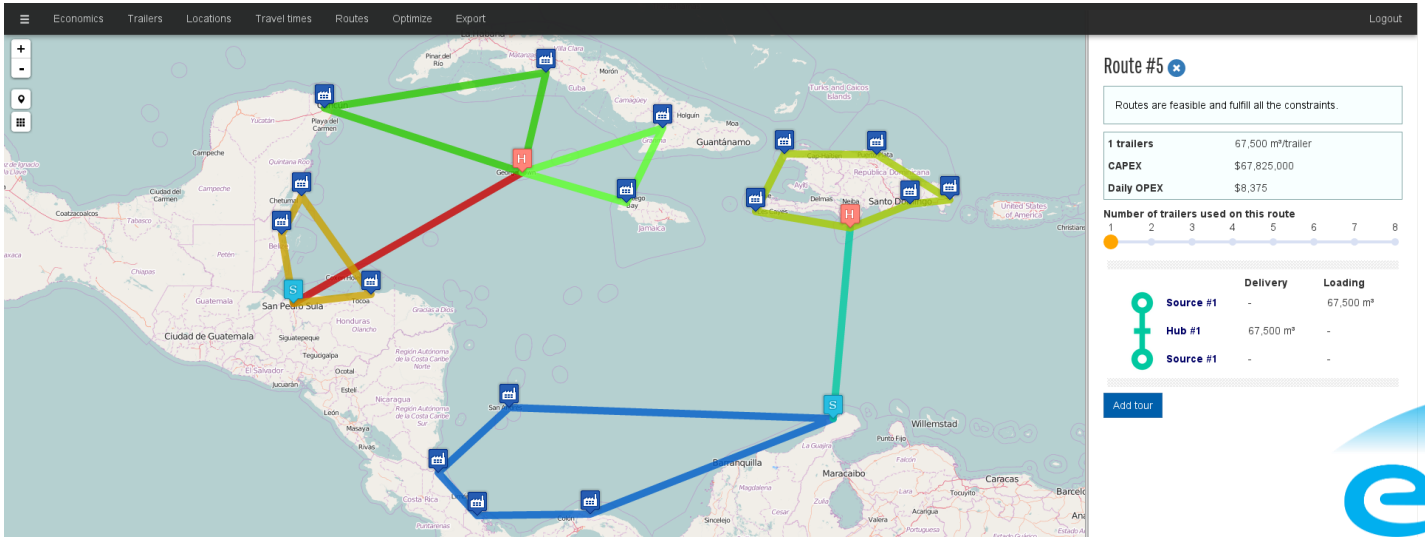
```
function model() {
  x[1..K] <- list(N) ; // for each vehicle, order the clients to visit
  constraint partition( x[1..K] ); // each client is visited once
  distances[k in 1..K] <- sum[i in 1..N-1]( dist( x[k][i-1], x[k][i]) )
                          + dist( x[k][N-1], x[k][0] ); // traveled distance for each vehicle
  minimize sum[k in 1..K]( distances[k] ); // minimize total traveled distance
}
```



Real-life VRP

LNG supply chain design

- LNG routes: sources/hubs → hubs/clients
- Sizing trailers and client/hub storages
- Until 10 sources/hubs and 100 clients



Direct LocalSolver approach: no simplification, no decomposition

Beyond routing problems

Planning, sequencing, scheduling

Fichier Édition Affichage Historique Marque-pages Outils ?

Example tour — LocalSolv...

www.localsolver.com/documentation/exampletour/index.html

LocalSolver

Search docs

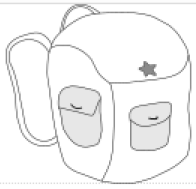
- Installation & licensing
- Quick start guide
- Advanced features
- LSP Reference Manual
- **Example tour**
  - Toy
  - Knapsack
  - P-median
  - Branin function
  - Optimal bucket
  - Smallest circle
  - Max cut
  - Social golfer
  - Car sequencing
  - Steel mill slab design
  - K-means
  - Travelling salesman problem
  - Quadratic assignment problem
  - Flowshop
  - Vehicule routing problem
- Python API Reference
- C++ API Reference
- Java API Reference

List variables

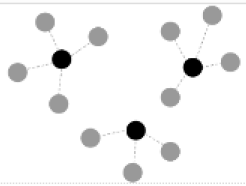
Docs » Example tour

Example tour

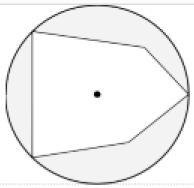
Toy ★



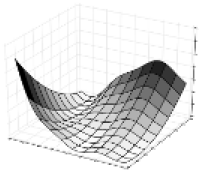
Knapsack ★



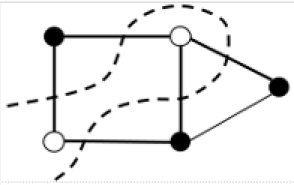
P-median ★



Smallest circle ★

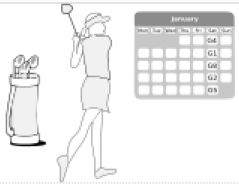


Branin function ★

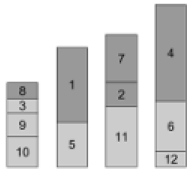


Max cut ★

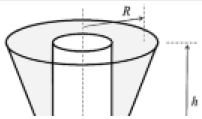
Car sequencing ★★

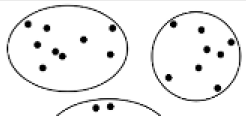


Social golfer ★★



Steel mill slab design ★★

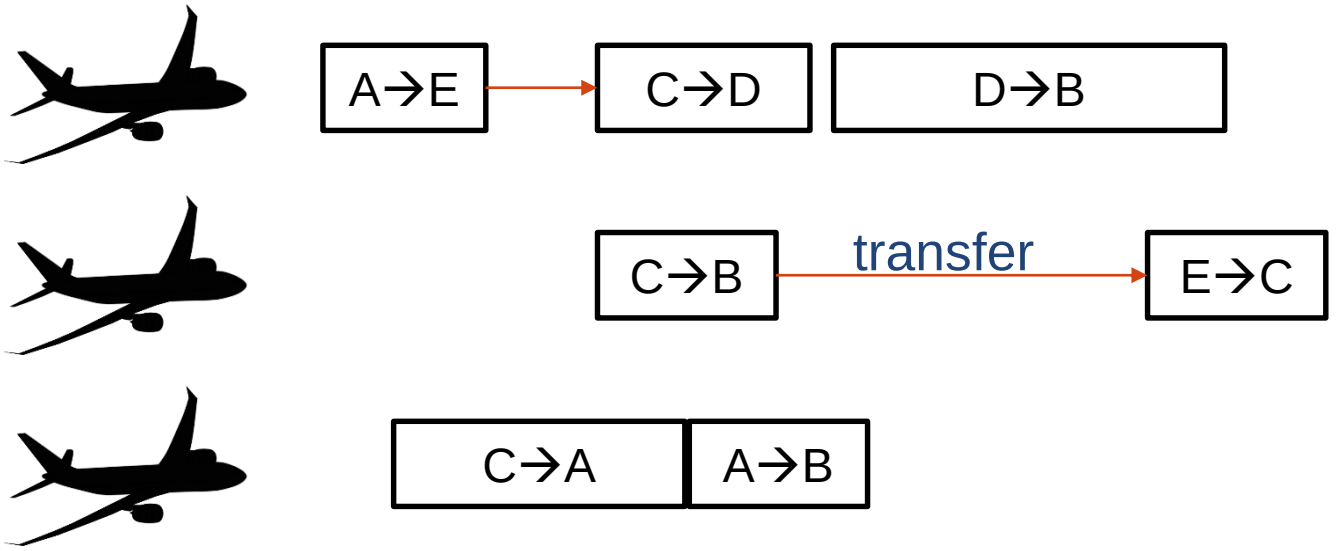




Uberising private jet business

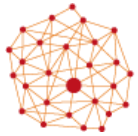
Flights to plane assignments

STELLAR
www.stellar.aero



A solution is a partition of flights into K lists (one per plane)

The goal is to minimize the total transfer times



LocalSolver

Solving routing and scheduling problems using LocalSolver

Set-based modeling in LocalSolver 6.5

www.localsolver.com

Frédéric GARDI
Co-Founder & Managing Partner
fgardi@localsolver.com